Available online at www.sciencedirect.com
SCIENCE DIRECT®

ELSEVIER

Computers & Operations Research 31 (2004) 201–222

computers & operations research

www.elsevier.com/locate/dsw

A queueing control model for retail services having back room operations and cross-trained workers

Oded Berman[a,*], Richard C. Larson[b]

[a] *Joseph L. Rotman School of Management, University of Toronto, 105 St George Street Toronto, Toronto, Ont., Canada M5S 3E6*
[b] *Operations Research Center, Massachusetts Institute of Technology, Cambridge, MA 02139, USA*

Received 1 March 2002; received in revised form 9 September 2002

Abstract

Many retail service facilities have both front room and back room operations. The front room deals with serving customers, perhaps from a queue. The back room focuses typically on restocking of shelves and sorting and/or processing of paperwork. We model such a facility having workers who are cross-trained to do both jobs. We assume that back room work is interruptible. A manager can bring a worker from the back room to the front room when the customer checkout queue becomes “too long”. A reverse assignment occurs when the number of customers is sufficiently small.

We assume that the retail facility contains two customer-counting technologies. The first counts the number of shoppers in the store who are not already in checkout queues and the second counts the number of customers at checkout. The goal is to find the minimum number of workers to staff the facility subject to two performance constraints. The mean queue delay in the front room must be less than a pre-specified value, and the time-average number of workers in the back room must be greater than a pre-specified value. Once the minimum complement of workers is found, a secondary goal is to minimize mean queueing delay subject to retaining back room feasibility. The system is a continuous time Markov process having a three-dimensional state space. The (heuristic) optimization process includes state-dependent rules for switching workers between rooms.

Scope and purpose

Many customer service facilities have both front room and back room operations for workers. In the front room, workers serve customers who may be delayed in a queue. In the back room workers perform required but time-insensitive work such as restocking of shelves, sorting of materials and/or processing of paperwork. Allowing workers to be cross-trained to perform both functions creates a potentially high productivity system with worker utilization rates near 100%. We model such a facility having workers who are cross-trained using

[*] Corresponding author. Tel.: +1-416-978-4239; fax: +1-416-978-5433.
E-mail address: berman@rotman.utoronto.ca (O. Berman).

PII: S0305-0548(02)00180-6

techniques of queueing theory and heuristic optimization. The assignment of workers over time to each of the two tasks is dependent on the number of customers in the front room. When the customer queue becomes "too long", the manager can bring a worker from the back room to front room and vice versa when there are few customers in the front room. The goal is to find the minimum number of workers to staff the facility subject to performance constraints in both rooms: a mean queue delay constraint in the front room and an average number of workers constraint in the back room. Of particular interest is determining the rules for bringing in and sending back workers.

Keywords: Queueing model; Control model; Backroom operations; Cross-trained workers; Heuristics; Switching rules

1. Introduction and overview

We consider a problem motivated by retail stores and their need to provide good service to customers and simultaneously to reduce costs. The general problem is as follows: For *line workers* in the store there are basically two types of work, serving retail customers in the *front room* and performing *back room* tasks or chores. The back room work is much less time sensitive than the servicing of waiting customers in the front room. The work in the back room is interruptible, allowing a worker in the back room to switch to front room work with little delay or lost productivity. The switch from back room to front room would occur at those moments when the queue of waiting customers in the front room becomes "too long". The reverse switch from the front room to the back room occurs once the number of retail customers is sufficiently small. Returning to the back room from the front room may require some "re-orientation time" on the part of the switched worker. This is due to the fact that some or all of the back room work may be task-oriented requiring concentrated work in sorting, stocking, counting, etc. It may take a few minutes for the worker to acclimate to the status and necessary next steps of the back room work. Thus the disruption caused by temporary assignment to the front room may cause a delay in time until the worker re-establishes full productivity.

Operations of the United States Postal Service (USPS) motivated our work on this problem. The USPS is establishing a national network of "retail postal stores". These stores will have up to three types of workers: retail-only service professionals, backroom only workers, and cross-trained "switchers" who can do both front room and backroom work. Rational deployment of the switchers is the focus of our work. Subsequent to the USPS work, other large retail firms and also hotels have expressed interest in this approach.

The proposed system that manages the switching of workers utilizes real-time information about the numbers of customers present in two different areas in the store. The first is the customer count in the front room, or at "customer checkout", customers either in service or in queue. The second is the count of customers who are elsewhere in the store as "shoppers" who are not yet at checkout queues. These two different real-time customer counts could be derived from any of a variety of monitoring technologies. For instance, slow scan television that transforms human images to stick figures and then "counts them" could be used to monitor the number of customers at checkout. Or, "electronic eyes" at the entrances to the store and entrances to the checkout area could provide the count of shoppers not yet at checkout. The Disney company uses such technology to maintain time-of-day counts of “guests” who enter and leave the various attractions at Disney theme parks. The information is relevant because it can be used to anticipate near-term flows to the checkout queue, thereby allowing workers to pre-position themselves more effectively. For instance, if there are many shoppers in the store even when the checkout queues are not long, the best policy may retain a greater than average number of workers “up front” to anticipate a likely near term “surge” of customers at checkout.

The workforce optimization problem is to minimize the total number of workers in the store subject to two performance constraints: a “level of queueing service” constraint in the front room and a constraint on the time-average number of Full Time Equivalent (“FTE”) workers present in the back room. An important component of finding the least cost solution is optimizing the rules upon which the room switching is based.

Almost all of us have experienced this type of queueing system in our daily lives. We go into a retail establishment, get our items for purchase, walk to the check out area, and observe a queue longer than we would like it to be. If we are lucky eventually the store manager calls up one or more workers from the back of the store to open up additional check out registers. We doubt that many retail establishments attempt to create optimal or nearly optimal decision rules for worker switching based on the number of customers at checkout *and* the number of shoppers in the store.

We model the queue as a continuous time Markov process operating in steady state and offer a heuristic for its optimization. The paper builds from a rich tradition in the theory of queues dating back to 1957, where the focus is multi-server queues with a variable number of servers. The literature review is delayed until Section 5 of the paper, after the model is developed and the optimization problem defined.

2. The model

In this section we define the model and present key formulas.

Define

$w =$ total workers in facility, front and back (w positive integer),

$\bar{B} =$ time−average number of workers in back room,

$\bar{F} =$ time−average number of workers in front room,

$\bar{B} + \bar{F} = w$,

$w_f =$ Number of workers permanently assigned to the front room (clearly $w_f \leqslant \bar{F}$).

Each time a worker returns to the back room, she loses τ ($\tau \geqslant 0$) hours in a “re-adjustment phase”, For simplicity, we assume that this back room work time is lost in its entirety even if the worker or another worker is pulled out of the back room again before the τ time units are completed. If v is the frequency of worker returns to the back room per hour, then the effective time-average worker complement in the back room is

$$\bar{B}_\mathrm{e} = \bar{B} - v\tau. \tag{1}$$

Eq. (1) has an interesting physical interpretation. Each worker returns to the back room may be considered to be a “queueing system arrival”, at arrival rate v and sojourn time τ, where those in this queue are becoming re-oriented to do back room work. Little’s Law [1] applied to this queue states that the time-average number of back room workers in this queue is $v\tau$. Hence the time-average number of back room servers "left over" to do real back room work is $\bar{B}_{\mathrm{e}} = \bar{B} - v\tau$. If τ or v is sufficiently large, $\bar{B}_{\mathrm{e}}$ could be negative! In such a case there would be no steady state and the queue of "fictional customers" would grow without bound. Feasibility requirements in our problem preclude this potential difficulty.

Retail customers arrive to the store entrance(s) in a Poisson manner with rate λ customers/hour. Once a customer enters the store (and is "counted" by the customer monitoring technology) he requires a length of time to "shop" before going to the checkout area. The population of all such shopping customers comprises one of three components of the system state variable. We assume that shoppers take exponentially distributed shopping times with mean γ^{-1}. Services at checkout are done independently with exponential service times, with mean service time equal to μ^{-1}.

Shopping times and checkout times are assumed to be mutually independent. This assumption is clearly an approximation but one that is nearly always made in tandem queues. On the one hand, one could argue that a shopper who spends a longer time shopping should also require a longer time at checkout because she is likely to have accumulated more items. On the other, much of shopping is travelling down the rows of the store, looking and deciding . The actual act of placing an item in a cart may represent a small fraction of the total shopping time. Moreover, at checkout portions of the service time may be unrelated to the number of items purchased, such as use of coupons and/or credit cards. Thus, we believe that our assumption of mutual independence of shopping times and checkout times is much better than say in digital communication systems, in which the criticism of such an assumption is often justified: the same number of bytes is involved at each queue station.

We assume that customer arrival-rates, store shopping times and shopping floor space are such that we can model the store-shopping queue as an $M/M/\infty$ queue. For computational tractability we 'clip' the number of possible customers in the queue (i.e., the number of shoppers in the store and not at checkout) at a very large but finite number C, so large that the chance of a customer being turned away from the store is very small. Our control policies do not affect the numerical value of the small fraction of customers who are turned away. (In all of our formulas involving the physics of operations, we deliberately ignore any impact of the finite state space, as the truncation level is set to have no measurable impact.)

A triplet of non-negative integers gives the state of the system at any time,

$(i,j,k) =$ system state having i servers in the front room, j customers in the checkout system (both in service and in queue), and k pre-checkout shoppers in the store. $i = w_f, w_f + 1, \ldots, w$; $j = 0, 1, 2, \ldots$; $k = 0, 1, 2, \ldots, C$.

In the (i,j,k) state space, probabilistic flows tend to travel in three-dimensional (3-D) loops or cycles. This behavior is distinctly different from the more familiar birth and death queues found in queueing textbooks. A typical loop may start near the origin with the system nearly empty, i.e., indices j,k at or near zero and index i at w_f. Then as more customers enter the store and eventually join the queue at checkout, the state of the system moves positively, first in the k direction and then in the j direction. Eventually, the queue manager brings a server from the back room, perhaps twice or even three times. As these servers are added, the state of the system then moves positively in the i direction. Eventually, the shopper population and then the checkout queue subside, moving the state toward lower values of k and j, and ultimately the queue manager returns one or more workers to the back room. As the number of servers decreases, the system state tends back toward low values of all three indices, toward system states having few customers and few servers. This cycle repeats continuously.

2.1. Performance constraints

We assume that the system must operate in accordance with stated management performance objectives for both checkout and the back room. For checkout we seek to limit the customer waiting time in queue to within some pre-specified limit. For the back room, we must have sufficiently many workers there (on average) so that all the back room work can be completed.

To accomplish these performance objectives, we first require that the mean queueing delay for a random customer denoted by $\bar{W}_q$ not exceed a given threshold T_0

$$\bar{W}_q \leqslant T_0. \tag{2}$$

Second, we require that the effective time-average worker complement in the back room must equal or exceed some minimum specified threshold

$$\bar{B}_e \geqslant B_0, \tag{3}$$

where

$B_0 =$ minimum time-average manpower complement to complete all the back room work. (This is expressed in FTE’s number of back room workers).

We do not explicitly model any workers who are permanently assigned to the back room. If there are such workers deployed, we assume that B_0 is accordingly reduced so that the “switchers” must satisfy Eq. (3).

2.2. Decisions to be made

We define two sets of worker room switch thresholds: the “upper thresholds”, used for determining when we move workers from the back room to the checkout area, and “lower thresholds”, used for the reverse assignment. For simplicity, we consider policies in which (i) worker movements can occur only upon either entrance of a new customer at the checkout queue or completion of service of a customer at checkout (the former allows a worker to be switched from the back room to the front; the latter allows the reverse); (ii) only one worker may be switched at a time.

We want to determine optimal values of w and of the thresholds for moving workers from one room to another. The optimal value of w will be its minimum value subject to constraints 2 and 3. This is a least cost solution, assuming that all workers have the same hourly pay rate. Once this minimum value is found we want to continue optimizing the transition thresholds in order to minimize $\bar{W}_q$ subject to satisfying $\bar{B}_e \geqslant B_0$.

2.3. The worker room switch thresholds

Notice that even though switching policies do not allow worker switches at moments of new shopper arrivals to the store entrance, the queue manager takes into account the number of shoppers in the store when worker switches are made.

Define the upper thresholds as

U_{ik}: given that there are k shoppers in the store (not at checkout) and i workers staffing checkout, U_{ik} is the trigger number of customers at checkout (in queue and in service) such that, whenever the system transitions from U_{ik} to $U_{ik}+1$ customers at checkout, the number of active front room servers instantaneously increases from i to $i+1$. $i = w_f,\ w_f+1,\dots,w-1$; $k = 0,1,2,3,\dots,C$.

Define the lower thresholds as

L_{ik}: given that there are k shoppers in the store (not at checkout) and i workers staffing checkout, L_{ik} is the trigger number of customers at checkout such that whenever the system transitions from L_{ik} to $L_{ik}-1$ customers at checkout, the total number of active front room workers instantaneously decreases from i to $i-1$, $i = w_f+1,\dots,w$.

If there exists one or more values of i for which there may be no further reduction in front room staffing, we denote for those values of i only $L_{ik}=0$. Define the vectors, $\underline{U}_k = (U_{w_f,k}, U_{w_f+1,k},\dots, U_{wk}), \underline{L}_k = (L_{w_f,k}, L_{w_f+1,k},\dots,L_{wk})$. Obviously $U_{ik} \geqslant L_{ik}$ for any $i = w_f,\dots,w$.

3. Queueing physics

In this section we develop the required Markovian balance equations for a specific set of thresholds. For transitions that involve moving servers, the balance equations represent flows "after the action" of having moved the server.

3.1. Balance of flow equations

Define

P_{ijk} = steady-state probability of the system being in state (i,j,k)

We construct the set of simultaneous linear equations whose solution yields the numerical values for the desired steady-state probabilities. These "balance of flow" equations require input flow to equal output flow for each state. In the following discussion, to simplify the writing of equations, we use a bracket "1{}" notation as follows:

1{*set of inequalities and equalities*} is an indicative binary variable that is equal to 0 or 1.

The general detailed balance equation that takes into consideration all possible transitions can be written as follows:

$$
\begin{aligned}
&(1\{k < C\}\lambda + \min(i,j)\mu + k\gamma)P_{ijk} \\
&\quad = 1\{j > L_{ik} - 1\}\min(i,j+1)\mu P_{i,j+1,k} + 1\{k > 0\}\lambda P_{i,j,k-1} \\
&\qquad + 1\{k < C \text{ and } j < U_{i,k+1} + 1\}(k+1)\gamma P_{i,j-1,k+1}
\end{aligned}
$$

$$+1\{i < w \text{ and } j = L_{i+1,k} - 1\}\min(i+1, j+1)\mu P_{i+1,j+1,k}$$
$$+1\{i \geqslant w_f + 1 \text{ and } k < C \text{ and } j = U_{i-1,k+1} + 1\}$$
$$\times(k+1)\gamma P_{i-1,j-1,k+1}. \tag{4}$$

This equation holds for any $i = w_f, \dots, w,\ k = 0, 1, \dots, C, j = 0, \dots, U'_{ik}$ where $U'_{ik} = \max_j\{U_{ij}\}$. Eq. (4) takes into consideration the fact that for a transition to state (i, j, k) from $(i+1, j+1, k)$ to occur the number of customers in the queue j must be equal to $L_{i+1,k} - 1$ [otherwise a transition will occur from state $(i, j+1, k)$]. Also for a transition to occur to state (i, j, k) from $(i-1, j-1, k+1), j$ must be equal to $U_{i-1,k+1} + 1$ [otherwise a transition will occur from state $(i, j-1, k+1)$].

3.2. Deriving desired quantities

We now use Little's Law to derive required quantities. Using

$$\bar{L} = \lambda\bar{W}, \tag{5a}$$

$$\bar{W} = \bar{W}_\mathrm{q} + 1/\mu, \tag{5b}$$

$$\bar{W}_\mathrm{q} = (\bar{L}/\lambda) - 1/\mu, \tag{5c}$$

where $\bar{L}$ and $\bar{W}$ are, respectively, the time-average number of customers in the system and the mean waiting time in the system, we obtain the key front room performance measure,

$$\bar{W}_\mathrm{q} = \frac{1}{\lambda}\sum_{j=1}^{\infty} jP_{\bullet j \bullet} - \frac{1}{\mu}, \tag{6}$$

where the "dot" ($\bullet$) notation signifies summation over all values of the missing index.

For our problem we also have

$$\bar{B} = w - \bar{F} = \text{time-average number of workers in the back room}$$
$$\text{(ignoring lost productivity due to switching friction)},$$

$$\bar{B}_\mathrm{e} = \bar{B} - v\tau = \left(w - \bar{F}\right) - v\tau$$

or,

$$\bar{B}_\mathrm{e} = w - \sum_{i=w_f}^{w} iP_{i\bullet\bullet} - v\tau. \tag{7}$$

We need to compute v, the frequency of worker switching from back room to front room. State (i, U_{ik}, k) is the state having i front room workers and k shoppers for which any entrance to the queue brings another worker immediately from the back room to the front room. Similarly, (i, L_{ik}, k) is the state that sends a worker from the front room to the back room upon completion of service of a customer. Using this terminology we can write

$$v = \sum_{i=w_f}^{w-1}\sum_{k=1}^{C} k\gamma P_{i,U_{ik},k} = \sum_{i=w_f+1}^{w}\sum_{k=1}^{C}\min(i, L_{ik})\mu P_{i,L_{ik},k}. \tag{8}$$

So, finally in terms of steady-state probabilities and given model parameters, we obtain an expression for the key back room quantity,

$$\bar{B}_\mathrm{e} = w - \sum_{i=w_f}^{w} iP_{i\bullet\bullet} - \gamma\tau \sum_{i=w_f}^{w-1}\sum_{k=1}^{C} kP_{i,U_{ik},k}. \tag{9}$$

4. Optimization problem

To find a best solution, we first want to find a feasible solution having the least number of workers w in the system. We expect that several perhaps many possible values for the threshold vectors will yield the same least cost worker complement. If that is the case, then after we find the least cost worker complement, we want to continue to reduce mean queueing delay in the front room, subject to continued back room feasibility.

We formulate the problem in two parts, A and B. In Part (A) we find the smallest complement of workers such that: (1) the mean queue delay will not exceed T_0, (2) the effective worker complement in the back room will not be below $B_{0,}$ and (3) all detailed queue balance equations hold

(A) $\mathrm{Min}\, w = w^*$

s.t. (2), (3) and s.t. the queue balance equations with values for $\underline{U}_k$ and $\underline{L}_k$ (vectors of lower and upper threshold defined earlier) yielding a feasible solution with w^* for all $k = 0, 1, 2, 3, \ldots$.

In Part (B) we try to find the smallest possible mean queue delay (which is at most T_0 from Part (A)) by finding the optimal thresholds.

(B) $\mathrm{Min}\, (\bar{W}_\mathrm{q} | w = w^*)$

s.t. (2), (3) and s.t. the queue balance equations with "optimal" values for $\underline{U}_k$ and $\underline{L}_k$, for all $k = 0, 1, 2, 3, \ldots$.

5. Relevant literature

Our work falls under the broad category of optimal control of queues having a variable number of servers. Our model is a 3-D generalization of the works of Romani [2] and Moder and Phillips [3]. In both papers the state variable has two components, the number of active servers and the number of customers in the system (in queue and in service). Romani considers a system in which a new server is added each time the queue length grows to a given threshold value; servers are released when a server completes service on a customer and there are no customers in queue. Moder and Phillips generalize Romani's model by assuming that there are a fixed number of servers always assigned to service regardless of the number of customers present and that the total number of servers available is bounded. Also, in the Moder–Phillips model, the thresholds at which servers are removed from service are more general than in Romani's paper. Both papers are descriptive, in that no optimization of server switches is attempted, but rather their movements are pre-described. Formulas for key quantities are obtained.

Yadin and Naor [4] further generalize the above two models. They assume that a finite set of service rates is available to a queue controller. The problem is determining the conditions under which a particular service rate should be selected. Suppose that the feasible service rates are $\mu_0, \mu_1, \ldots, \mu_k, \ldots$, where $\mu_{k+1} > \mu_k$ and $\mu_0 = 0$. The queue control policy is then stated: "whenever system size reaches a value R_k (from below) and service capacity equals μ_{k-1}, the latter is increased to μ_k; whenever system size drops to S_k (from above) and service capacity is μ_{k+1}, the latter is decreased to μ_k". For any given control policy (specified by vectors $\{R_k\}$ and $\{S_k\}$), key performance measures are derived, including steady state probabilities, mean queue length and the rate v of server switches. Note that the Yadin–Naor model is essentially a 2-D version of our model, where our service rates are integer multiples of μ_1, and the optimization problem is finding the optimal vectors $\{R_k\}$ and $\{S_k\}$ under a given cost structure. Yadin and Naor point out the extreme difficulty in finding such an optimal policy.

Serfozo [5] and Serfozo and Lu [6] prove that under certain conditions the optimal policy for models such as that of Yadin and Naor are monotone hysteretic policies, and that the resulting 2-D state space has a hysteresis loop of flows. Hysteresis is a lagging of an effect behind its cause. Hysteresis occurs in the control of these types of queueing systems when a cost is imposed on each switch of a server. If there were no cost of switching, then servers would be brought in instantaneously on a just-in-time basis, a truly simplistic optimal policy.

None of the cited researchers attempted to optimize the server switching process, model back room constraints, or minimize the total manpower w. And none included a third dimension in the state space representing a staged or tandem queueing process.

Other research in the optimal control of queues includes a great deal of work on $M/G/1$ queues and in particular servers who are switched on and off ("bang bang control"). A good synopsis of this work is in Gross and Harris [7, pp. 308–312].

6. Optimization heuristic

Our model is a tandem queue, with the first of two queues in series being the $M/M/\infty$ queue (assuming C is very large). The second queue in tandem is a variation of the hysteresis 2-D queue of Yadin and Naor [4], Serfozo [5] and Serfozo and Lu [6]. The steady-state output process of an $M/M/\infty$ queue is a Poisson process. Thus, if the server switch control policy ignored the number of customers in the $M/M/\infty$ queue (i.e., the number of shoppers in the store), then the second queue would be identical to those 2-D queues. One might be tempted to think that the steady-state probabilities of the entire two-stage (3-D) system could be obtained by the product of the steady-state probabilities of the two individual queues. While it is true that the steady-state number of customers in the $M/M/\infty$ queue has a Poisson distribution (see Gross and Harris [7]), the numbers of customers and active servers in the second queue are dependent on the state of the $M/M/\infty$ queue. Conditioned on the state of the $M/M/\infty$ queue, the second queue is driven by a modulated Poisson process whose instantaneous rate parameter is directly proportional to the number of (real-time) customers in the $M/M/\infty$ queue, and the configuration of the second queue is dependent on its current conditional arrival rate. The two queues are not independent and their steady state probabilities are not obtained by the multiplication of two marginal probability distributions.

Our task is to find optimal or near optimal values for the two "control matrices"

$$U=\left\{\begin{array}{cccc} U_{w_f,0} & U_{w_f,1} & \dots & U_{w_f,C} \\ U_{w_f+1,0} & U_{w_f+1,1} & \dots & U_{w_f+1,C} \\ \vdots & & & \\ U_{w0} & U_{w1} & \dots & U_{wC} \end{array}\right\}, \quad L=\left\{\begin{array}{cccc} L_{w_f,0} & L_{w_f,1} & \dots & L_{w_f,C} \\ L_{w_f+1,0} & L_{w_f+1,1} & \dots & L_{w_f+1,C} \\ \vdots & & & \\ L_{w0} & L_{w1} & \dots & L_{wC} \end{array}\right\}$$

with rows $i=w_{\mathrm{f}},\dots,w$ (number of servers) and columns $k=0,\dots,C$ (number of customers in the store). Optimal here means minimizing total manpower w subject to (2) and (3) and the queueing balance-of-flow equations (4). We do not have a provably optimal algorithm but rather a heuristic that has been shown to yield very good solutions. We choose not to utilize an approach based on Markov decision processes, due in part to the huge number of states that would have to be included and in part to the degenerate nature of the optimal solution (i.e., switching a server is an all or none decision).

The general approach is to attempt first to optimize the 2-D version of the problem, obtained naturally within our framework by letting the mean customer shopping time $(1/\gamma)\to 0$ or equivalently $\gamma\to\infty$. The best solution for the 2-D case is then imposed on the 3-D model, and successively improved by partitioning the 3-D state space along the k-axis. The process begins with a two-level partitioning at the median of the Poisson distribution for k, and subsequently divides the state space at iteration n into $n+1$ regions. A separate server control policy is found for each element of the partitioning. The partitioning process stops when no further significant improvement is found.

Computationally, rather than seek closed form solutions to the 3-D system of equations, we use the fast-converging Gauss–Siedel procedure discussed in Berman and Larson [8].

6.1. The 2-D heuristic

Our 2-D heuristic is motivated by the hysteresis optimality results of Serfozo [5] and Serfozo and Lu [6] and by management ease of implementation. In the 2-D problem the upper and lower thresholds for server switching are defined as functions of the number of servers only, U_i and L_i, as the number of shoppers k is suppressed. Also to make the presentation of the 2-D (and the 3-D) heuristic more readable we assume without loss of generality that $w_f=1$.

The following five conjectures, all assumed to apply in an optimal server control policy, are used in our heuristic to limit the set of thresholds that will be considered:

Conjecture 1. Decreasing an upper threshold reduces mean queue wait but also decreases time-average back room manpower.

Conjecture 2. The upper and lower thresholds increase monotonically with i, the number of servers in the front room (hysteresis effect).

Conjecture 3. Servers do not switch from front room to back room in the presence of queued customers nor from back room to front room without waiting customers.

Conjecture 4. Decreasing lower thresholds decreases mean queue delay.

Conjecture 5. If for any $\underline{L}=(L_1,L_2,\dots,L_w)$ there exists $\underline{U}=(U_1,U_2,\dots,U_w)$ such that $\bar{W}_{\mathrm{q}}(w,\underline{L},\underline{U}) > T_0$ and $\bar{B}_{\mathrm{e}}(w,\underline{L},\underline{U}) < B_0$, then there exists no feasible solution for that value of $\underline{L}$.

Conjecture 1 states that a policy that pulls workers from the back room at a smaller threshold queue length will result in reduced mean queueing delay but also less time-averaged manpower in the back room. Conjecture 2 is motivated by the hysteresis results of Serfozo [5] and Serfozo and Lu [6]. Conjecture 3, in addition to being mathematically intuitive, can be also considered as a reasonable management policy. If a server switched from front room to back room in the presence of a queued customer, the waiting customer could reasonably believe that management values back room work more highly than serving customers. Our policy avoids such customer perceptions. But we also believe the policy to be mathematically optimal, as placing a queued customer immediately into service provides for reduction in the length of the current busy period and may remove the need to switch back to the front room again (with penalty) during the current busy period. Conjecture 4 states that, allowing servers to stay longer in the front room before switching to the back room provides better queue performance. Decreasing lower thresholds has a complicated effect on the time-average back room manpower. The direct effect is that servers will stay longer in the front room and thus the back room will "get worse" in terms of time-average back room manpower. However, for some lower thresholds (particularly small ones) the result can be servers entering and leaving the back room with a smaller frequency v, thereby reducing friction, and thus the overall effect on the back room may be beneficial [refer to Eq. (1)]. This is taken into account in the 2-D heuristic where a local maximum of mean back room manpower is sought when decreasing the lower threshold values. Conjecture 5 can be argued intuitively as follows: for fixed $\underline{L}$ changing any component of $\underline{U}$, say U_i, by ± 1 improves one performance measure at the expense of the other. That is, if subtracting 1 from U_i makes $\bar{W}_{\mathrm{q}}$ feasible, it makes $\bar{B}_{\mathrm{e}}$ even further from feasibility. The reverse is true if one adds 1.0 to U_i : $\bar{B}_{\mathrm{e}}$ may become feasible but $\bar{W}_{\mathrm{q}}$ is further from feasibility. Thus there will not be any feasible solution for any $\underline{L}$. The argument is not rigorous as multiple switches, say adding 1 to U_i and subtracting 1 from U_{i+1}, are not ruled out by this argument.

The 2-D heuristic has two stages (Fig. 1), reflecting parts (A) and (B) of the optimization problem. Recall the first stage is to find a feasible solution having the smallest total manpower, i.e., $w=w^*$. The second is to find for w^* the smallest possible $\bar{W}_{\mathrm{q}}$ subject to $\bar{B}_{\mathrm{e}} \geqslant B_0$. At the end of Stage 1, there is no possibility to reduce w further without violating either $\bar{W}_{\mathrm{q}}$ feasibility or $\bar{B}_{\mathrm{e}}$ feasibility or both. In Stage 2 there is an effort to try to take advantage of any slack in the $\bar{B}_{\mathrm{e}}$ constraint and to reduce $\bar{W}_{\mathrm{q}}$ by decreasing this slack.

Stage 1

Here are the main components of Stage 1.

(1) Initialization

We start the first stage (top of Fig. 1) by treating the front room and back room as separate non-interacting facilities and staff each at the minimum possible staffing level:

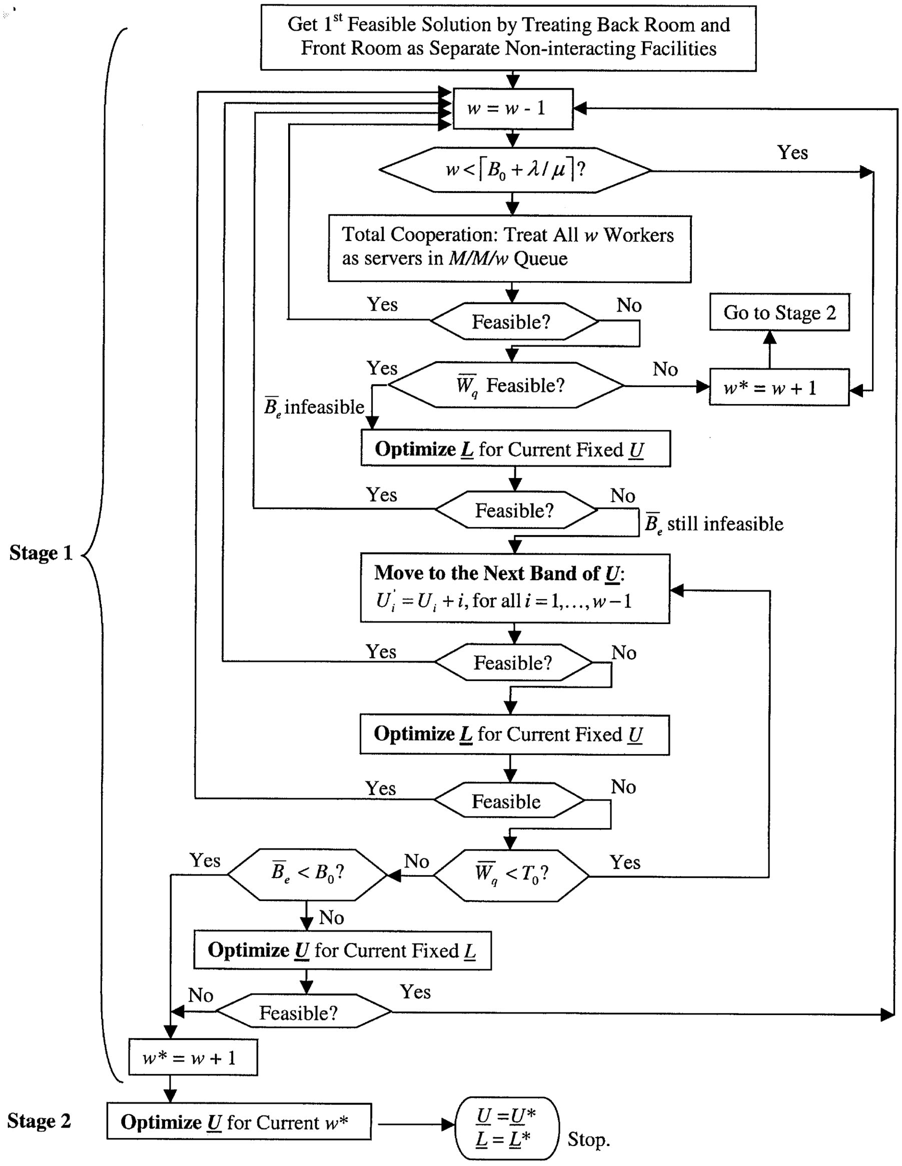
Get 1st Feasible Solution by Treating Back Room and Front Room as Separate Non-interacting Facilities
w = w - 1
w < ⌈B0 + λ/μ⌉?
Yes
Total Cooperation: Treat All w Workers as servers in M/M/w Queue
Yes
Feasible?
No
Go to Stage 2
Yes
W̄q Feasible?
No
w* = w + 1
B̄e infeasible
Optimize L for Current Fixed U
Yes
Feasible?
No
B̄e still infeasible
Move to the Next Band of U:
U'i = Ui + i, for all i = 1,...,w-1
Yes
Feasible?
No
Optimize L for Current Fixed U
Yes
Feasible
No
Yes
B̄e < B0?
No
W̄q < T0?
Yes
No
Optimize U for Current Fixed L
No
Feasible?
Yes
w* = w + 1
Stage 1
Stage 2
Optimize U for Current w*
U = U*
L = L*
Stop.

Fig. 1. The suggested 2-D optimization algorithm.

Back room: $b = \lceil B_0 \rceil$, where $\lceil x \rceil \equiv$ smallest integer greater than or equal to x.
Front room: Find smallest integer f such that $\bar{W}_q(f) \leqslant T_0$, where $\bar{W}_q(f)$ is the mean queue delay in the corresponding $M/M/f$ queue.

$$w \equiv b + f.$$

(2) Reduce the staffing level by one and test $M/M/w$ queue

Next we reduce the total staffing level by one and unless there is no steady-state solution ($w < \lceil B_0 + \lambda/\mu \rceil$), we test for feasibility with the $M/M/w$ queue. The $M/M/w$ queue represents total cooperation (meaning $L_1 = 0, U_1 = 1, L_w = w, U_w = \infty, L_i = U_i,\ i = 2, \ldots, w-1$) having minimum $\bar{W}_\mathrm{q}$ but maximum switching friction due to movements into and out of back room. Thus there is no way to improve $\bar{W}_\mathrm{q}$ given w, but $\bar{B}_\mathrm{e}$ may be improved by allowing fewer switches.

We repeat this feasibility loop using the $M/M/w$ queue until we first get to $\bar{W}_\mathrm{q}$ feasibility and $\bar{B}_\mathrm{e}$ infeasibility. If we find with the current proposed value of w that we do not even have $\bar{W}_\mathrm{q}$ feasibility, Stage 1 is finished and we go to Stage 2 with $w^* = w + 1$ servers.

(3) Working toward $\bar{B}_\mathrm{e}$ feasibility

Assume we are still in Stage 1 with $\bar{W}_\mathrm{q}$ feasibility and $\bar{B}_\mathrm{e}$ infeasibility. We work toward $\bar{B}_\mathrm{e}$ feasibility while maintaining $\bar{W}_\mathrm{q}$ feasibility. If we are successful at finding feasibility with w workers, we immediately return to the top of the loop and attempt the process again with w reduced by one. Eventually in this way we find the minimal value of $w, w = w^*$.

To move toward $\bar{B}_\mathrm{e}$ feasibility while maintaining $\bar{W}_\mathrm{q}$ feasibility, we use two main procedures: **Optimize $\underline{L}$** and **Move to the Next Band of $\underline{U}$**. With Optimize $\underline{L}$ we try to increase $\bar{B}_\mathrm{e}$ while not increasing (actually decreasing) $\bar{W}_\mathrm{q}$. **If Optimize $\underline{L}$** does not bring feasibility, the procedure **Move to the Next Band of $\underline{U}$** is used. With this procedure $\bar{B}_\mathrm{e}$ is increased but $\bar{W}_\mathrm{q}$ is increased as well.

In **Optimize $\underline{L}$**, for a given $\underline{U}$ a local maximum of $\bar{B}_\mathrm{e}$ is sought by sequential decreases of $\underline{L}$ component values. Recall that decreasing $\underline{L}$ decreases $\bar{W}_\mathrm{q}$ (Conjecture 4). We start by decreasing L_2 by one at a time until $L_2 = 0$ then L_3 and so on. As long as decreasing L_i results in larger $\bar{B}_\mathrm{e}$ (recall that decreasing $\underline{L}$ has a direct effect of decreasing $\bar{B}_\mathrm{e}$ and an indirect effect of increasing $\bar{B}_\mathrm{e}$ due to less switching friction) we continue. Once $\bar{B}_\mathrm{e}$ decreases, we retain in an incumbent solution the immediately previous value of L_i and continue with decreasing L_{i+1} as long as $i + 1 < w$ and so on.

In **Move to the Next Band of $\underline{U}$**, for a given $\underline{L}$, each U_i is increased by i for $i = 1, \ldots, w$. Increasing $\underline{U}$ increases both $\bar{W}_\mathrm{q}$ and $\bar{B}_\mathrm{e}$ (Conjecture 1). We note that based on our computational experience, **Move to the Next Band** was shown to give better results than several other tested procedures to increase the upper thresholds. The same is true for **Decrease $\underline{U}$** procedure (where the move is to the previous band) discussed later. Thus this procedure ensures improvement of $\bar{B}_\mathrm{e}$. The two procedures are performed in a sequential alternating manner until either both $\bar{B}_\mathrm{e}$ and $\bar{W}_\mathrm{q}$ are feasible, in which case we return to the top of the loop with w decreased by one, or $\bar{W}_\mathrm{q}$ is infeasible. When $\bar{W}_\mathrm{q}$ becomes infeasible (following **Optimize $\underline{L}$**), there are two possibilities: (i) if $\bar{B}_\mathrm{e}$ is also infeasible, we set $w^* = w + 1$ and we terminate Stage 1 and start Stage 2; (ii) $\bar{B}_\mathrm{e}$ is feasible in which case we implement a third procedure called **Optimize $\underline{U}$**.

Optimize $\underline{U}$ is similar to **Optimize $\underline{L}$**, except that we decrease by one at a time the components of $\underline{U}$. **Optimize $\underline{U}$** decreases both $\bar{W}_\mathrm{q}$ and $\bar{B}_\mathrm{e}$ (Conjecture 1). We start by decreasing U_1 by one unit at a time until $U_1 = 1$, then U_2 by one unit at a time until $U_2 = 2$, then U_3 and so on as long as decreasing U_i retains feasible $\bar{B}_\mathrm{e}$. Once $\bar{B}_\mathrm{e}$ falls under B_0, we retain as an incumbent solution the immediately previous U_i and as long as $i + 1 < w - 1$, start decreasing U_{i+1} and so on. If **Optimize $\underline{U}$** results in a feasible solution we return to the top of the loop with w decreased by one. Otherwise $w^* = w + 1$ and we initiate Stage 2.

Stage 2

Stage 2 starts with the optimal workforce w^* and with $\underline{U}(w^*)$, $\underline{L}(w^*)$-the $\underline{U}$ and $\underline{L}$ vectors obtained just before the algorithms attempted to find a feasible solution with $w^* - 1$. In Stage 2 the objective is to minimize $\bar{W}_{\mathrm{q}}$ while maintaining $\bar{B}_{\mathrm{e}}$ feasibility. **Optimize $\underline{U}$** is the main procedure used here.

6.2. The 3-D heuristic

For the 3-D model we add two additional conjectures that limit the set of possible thresholds over which the heuristic will search:

Conjecture A. The conjectures of the 2-D model hold in general for each $k=0,1,\ldots,C$. This implies that the components in columns of **L** and **U** (now matrices) are monotonically increasing.

Conjecture B. For each $i=1,\ldots,w$, $U_{i0} \geqslant U_{i1} \geqslant \cdots U_{iC}$ and $L_{i0} \geqslant L_{i1} \geqslant \cdots \geqslant L_{iC}$ (monotonicity in rows).

According to Conjecture B, as k the number of customers in the store increases, we require fewer customers in queue to bring up personnel from the back room and we will tend to retain servers in the front room under more stringent rules for returning to the back room. Call this the ‘short term anticipation’ conjecture, in which larger number of in-store shoppers who are not yet at checkout reduce our tendency to switch workers away from checkout and needlessly incurring switching penalty costs (as the switched workers would likely have to be re-switched from the back to the front room). Throughout execution of the heuristic we require that we do not violate the monotonicity of conjectures A and B.

A flow chart of the 3-D heuristic is included in Fig. 2. The heuristic starts with the minimum cost solution for the model with no cooperation (the front room and back room are separate non-interacting facilities). Then w is reduced by 1 and unless there is no steady state solution ($w < \lceil B_0 + \lambda/\mu \rceil$) a partial 2-D heuristic is solved for the current w value. The objective of the partial heuristic is to determine if there exists a feasible solution with the current w. It starts with (the fourth box from the top of Fig. 1) total cooperation and ends once a feasible solution is encountered or when the solution provided at the end of Stage 1 is infeasible.

If the partial 2-D heuristic finds a feasible solution w is reduced by 1, a flag is set to 1 ($FL=1$) and unless there is no steady state solution the partial 2-D is re-solved. Once the solution provided by the partial 2-D heuristic becomes infeasible for the current value of w, the 3-D part starts using this infeasible solution. If the 3-D part provides a feasible solution the flag is set to 0, the solution obtained is stored, w is reduced by 1 and unless there is no steady state solution the partial 2-D heuristic is re-solved. If the 3-D solution is infeasible, we either stop if the flag is equal to 0 (and then the stored solution is declared “optimal”) or w is increased by 1 and 2-D heuristic followed by the 3-D heuristic are re-solved once more. The reason why we stop this way is that the 2-D heuristic may not have performed fully (or not performed at all) with the value w^*. Thus re-solving (the complete) 2-D heuristic (that includes Stage 2) and the 3-D heuristic may provide a better solution.

We now discuss details of the 3-D heuristic. It starts with the (best) infeasible solution $(W, \underline{L}, \underline{U})$ obtained at the end of the partial 2-D heuristic (**Optimize $\underline{U}$** at the end of Stage 1 does not give a feasible solution). The initial matrices in 3-D heuristic L and U are of dimension $w \times C$ and contain

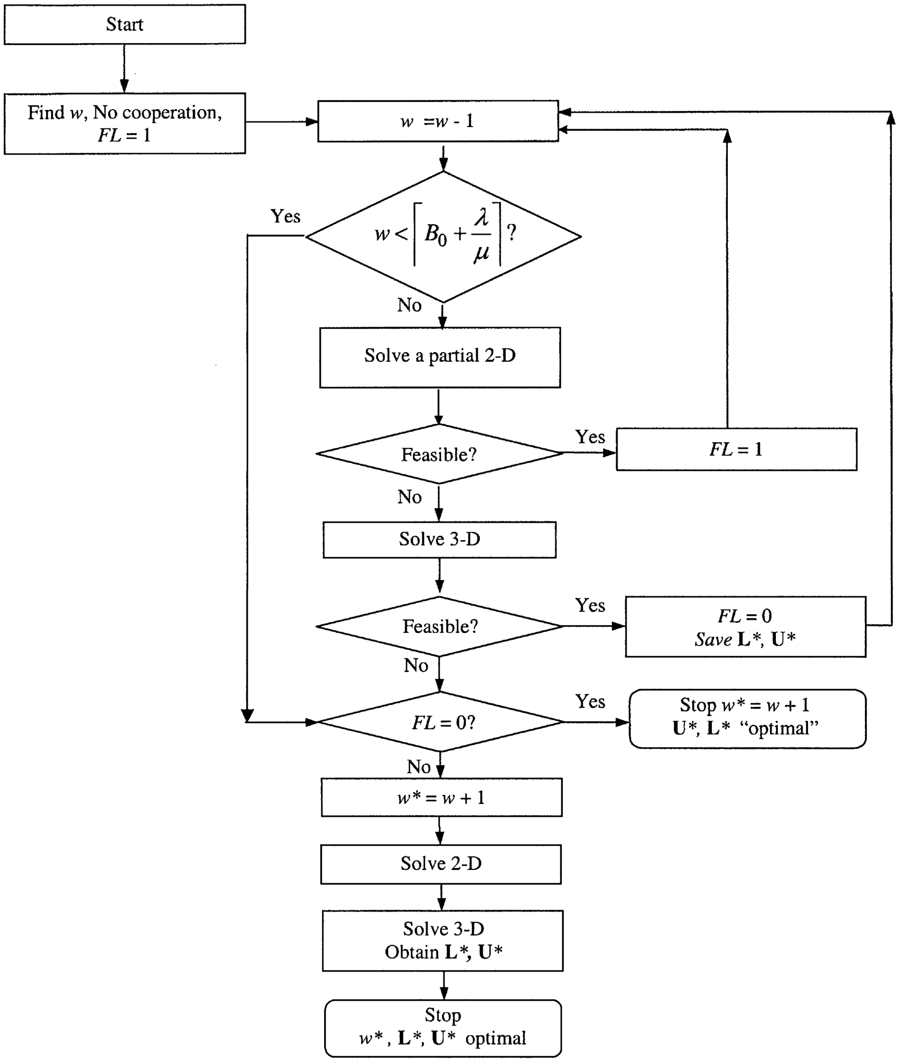
Start
Find w, No cooperation, $FL = 1$
$w = w - 1$
$w < \left\lceil B_0 + \frac{\lambda}{\mu} \right\rceil$?
Yes
No
Solve a partial 2-D
Feasible?
Yes
$FL = 1$
No
Solve 3-D
Feasible?
Yes
$FL = 0$ Save $\mathbf{L}^*$, $\mathbf{U}^*$
No
$FL = 0$?
Yes
Stop $w^* = w + 1$ $\mathbf{U}^*$, $\mathbf{L}^*$ "optimal"
No
$w^* = w + 1$
Solve 2-D
Solve 3-D Obtain $\mathbf{L}^*$, $\mathbf{U}^*$
Stop w^*, $\mathbf{L}^*$, $\mathbf{U}^*$ optimal

Fig. 2. A flow chart of the 3-D heuristic.

identical columns of $\underline{L}$ and $\underline{U}$, respectively. The 3-D part has 2 main procedures:

Decrease (Increase) *U*

For a given column k of U we move to the previous band, i.e. U_{ik} is reduced by i, for i=1,...,w−1 (increase-move to the next band, i.e., U_{ik} is increased by i for $i = 1,\dots,w-1$).

Decrease (Increase) *L*

For a given column k of L we decrease by 1 each L_{ik}, $i = 2,\dots,w$ (increase by 1 each L_{ik}, $i = 1,\dots,w$).

Decrease (Increase) *U* is essentially **Move to the Next (Previous) Band of $\underline{U}$** used in the 2-D heuristic, but now applied to a column (fixed number of customers at the store) of U. Decrease (Increase) L is essentially a step of **Optimize $\underline{L}$** applied to a column of L.

The 3-D part first calculates k^*, the median of the distribution of customers in the C-truncated $M/G/\infty$ model. For the current value of w the heuristic provides the "best" values of the thresholds L_{ik} and U_{ik} for $k \geqslant k^*$ and $k < k^*$. For $k \geqslant k^*$ it starts with $k = C$ and performs **Decrease *U***. Recall that **Decrease *U*** decreases both $\bar{W}_{\mathrm{q}}$ and $\bar{B}_{\mathrm{e}}$ and thus as long as $\bar{W}_{\mathrm{q}}$ is improved while $\bar{B}_{\mathrm{e}}$ remains feasible it continues with $k = C - 1$ and so on until there is no improvement or until $k = k^*$. Then **Decrease *L*** is performed starting again with $k = C$. As long as $\bar{W}_{\mathrm{q}}$ is improved while $\bar{B}_{\mathrm{e}}$ remains feasible k is decreased until reaching k^*. Then we repeat the **Decrease *U*/Decrease *L*** until for a full cycle there is no improvement. For $k < k^*$ we do the same except that we start with $k = 0$, increase k and perform **Increase *U*/Increase *L*** procedures. Note that to obtain $\bar{W}_{\mathrm{q}}$ and $\bar{B}_{\mathrm{e}}$ we must solve the balance equations of the model (for all k values).

We examine the solutions from the two halves of the distribution, corresponding to $k \geqslant k^*$ and $k < k^*$. If $k \geqslant k^*$ has yielded more improvement in reduction of $\bar{W}_{\mathrm{q}}$ (compared to the 2-D solution), we find the 0.75 fractile which we denote by k^{**} and repeat the process for both $k^* \leqslant k < k^{**}$ and $k^{**} \leqslant k \leqslant C$. If $k < k^*$ provides more improvement, we find the 0.25 fractile k^{**} and solve for $0 \leqslant k < k^{**}$ and $k^{**} \leqslant k < k^*$. We use $\underline{U}_{\mathrm{w}} = 25$ as a finite substitute for ∞. We continue with the process as long as there is an improvement (in reduction of $\bar{W}_{\mathrm{q}}$ while retaining feasibility in $\bar{B}_{\mathrm{e}}$).

6.3. Example

Consider the following example: $\lambda = 30$, $\mu = 20$, $\gamma = 10$, $T_0 = 0.05$, $B_0 = 3.13, \tau = 0.1$, and with one dedicated server in the front room. First we show the solution using just the 2-D heuristic. The algorithm starts with no cooperation between the front room and the back room and results in $w = 7$, $\bar{W}_{\mathrm{q}} = 0.00263$. Then w is reduced to 6 and with the partial 2-D the $M/M/6$ queue model ($\underline{L} = (0, 2, 3, 4, 5, 6)$, $\underline{U} = (1, 2, 3, 4, 5, \infty)$) gives $\bar{B}_{\mathrm{e}} = 1.9634$, $\bar{W}_{\mathrm{q}} = 0.00049$ which is not feasible. **Optimize $\underline{L}$** gives $\underline{L} = (0, 0, 2, 3, 4, 5)$, $\underline{U}$=$(1, 2, 3, 4, 5, \infty)$ with $\bar{B}_{\mathrm{e}}$=2.7394, $\bar{W}_{\mathrm{q}}$=0.00049. **Move to the Next Band** of $\underline{U}$($\underline{U}$=$(2, 4, 6, 8, 10, \infty)$) gives $\bar{B}_{\mathrm{e}}$=3.747, $\bar{W}_{\mathrm{q}}$=0.01795 which is feasible and thus w is reduced to 5 and the 2-D heuristic is re-solved. The 2-D includes three rounds of **Move to the Next Band** and a series of **Optimize *L*** and results with $\underline{L} = (0, 2, 3, 4, 5), \underline{U} = (4, 7, 9, 12, \infty), \bar{B}_{\mathrm{e}} = 3.1382$, $\bar{W}_{\mathrm{q}} = 0.0550$ (which is feasible in $\bar{B}_{\mathrm{e}}$ but not $\bar{W}_{\mathrm{q}}$). Threshold w is next increased to 6 and stage 2 starts with the best feasible solution of stage 1: $\underline{L} = (0, 0, 2, 3, 4, 5)$ and $\underline{U} = (2, 4, 6, 8, 10, \infty)$. Applying **Optimal *U*** results with the final solution $\underline{L} = (0, 0, 2, 3, 4, 5)$ and $\underline{U} = (1, 3, 4, 5, 6, \infty)$ with $\bar{B}_{\mathrm{e}} = 3.2644$ and $\bar{W}_{\mathrm{q}} = 0.00407$.

The 3-D heuristic is identical to the 2-D heuristic until we obtain the solution $w = 5$, $\underline{L} = (0, 2, 3, 4, 5)$, $\underline{U} = (4, 7, 9, 12, \infty)$ with $\bar{B}_{\mathrm{e}} = 3.1382$ and $\bar{W}_{\mathrm{q}} = 0.0550$ which is not feasible. Now instead of increase w to 6 as we did for the 2-D heuristic, the 3-D heuristic starts with the solution

$$\mathbf{U} = \begin{pmatrix} 4, & 4, & \ldots, & 4 \\ 7, & 7, & \ldots, & 7 \\ 9, & 9, & \ldots, & 9 \\ 12, & 12, & \ldots, & 12 \\ 25, & 25, & \ldots, & 25 \end{pmatrix}, \quad \mathbf{L} = \begin{pmatrix} 0, & 0, & \ldots, & 0 \\ 2, & 2, & \ldots, & 2 \\ 3, & 3, & \ldots, & 3 \\ 4, & 4, & \ldots, & 4 \\ 5, & 5, & \ldots, & 5 \end{pmatrix}.$$

(We used $C=10$ and $U_{wk}=25$ instead of ∞). The median of the truncated $M/G/10$ is 3 and $k \geqslant 3$ resulted in the feasible solution:

$$\mathbf{U}=\begin{pmatrix} 4 & 4 & 4 & 4 & 4 & 4 & 4 & 3 & 2 & 1 \\ 7 & 7 & 7 & 7 & 7 & 7 & 7 & 5 & 3 & 2 \\ 9 & 9 & 9 & 9 & 9 & 9 & 9 & 6 & 4 & 3 \\ 12 & 12 & 12 & 12 & 12 & 12 & 12 & 8 & 5 & 4 \\ 25 & 25 & 25 & 25 & 25 & 25 & 25 & 25 & 25 & 25 \end{pmatrix}$$

$$L=\begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2 & 2 & 2 & 2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 3 & 3 & 3 & 3 & 1 & 1 & 1 & 1 & 1 & 1 \\ 4 & 4 & 4 & 4 & 2 & 2 & 2 & 2 & 2 & 2 \\ 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 \end{pmatrix}$$

with $\bar{B}_{\mathrm{e}}=3.1307$, $\bar{W}_{\mathrm{q}}=0.0489$. A separate calculation using the 3-D heuristic for $k<3$ provided a worse solution. So, next we calculated the 0.75 fractile, which is 5. We find that both parts $3<k \leqslant 5$ and $5<k \leqslant 10$ do not improve the solution. So, we reduce w to 4, but since $\lceil 3.13+(30/20)\rceil = 5>4$, $w=5=w^*$ with the policy above is optimal.

6.4. Computational results

For the example discussed in the previous section, consider the following sets of parameter values: $\lambda=15,30,50,100$ and 200; $\tau=0.0,0.1,0.2,0.4,0.6,1.0$. Table 1 contains the results for all combinations of λ and τ values. The third column of the table contains the total number of workers assuming no cooperation—w^{NC}. The fourth column contains $w^{\min}$, the minimum possible number of workers before the system explodes. The fifth and sixth columns give, respectively, the optimal manpower $w^{\text{2-D}}$ and mean waiting time $\bar{W}_{\mathrm{q}}^{\text{2-D}}$obtained from the 2-D heuristic. The seventh column provides the percentage saving in manpower of the 2-D heuristic over the no-cooperation solution. The eighth and ninth columns contain the optimal manpower $w^{\text{3-D}}$ and mean waiting time $\bar{W}_{\mathrm{q}}^{\text{3-D}}$ obtained with the 3-D heuristic. The tenth column shows the percentage savings of the mean waiting time with the 3-D heuristic in contrast to the 2-D heuristic.

1. Notice that for $\lambda=30, \tau=0.1$, the 3-D heuristic improves by one the "optimal manpower" obtained with the 2-D heuristic (therefore we do not calculate the percentage savings in mean waiting time). Several conclusions can be drawn from Table 1:
   As expected, with no friction (i.e., $\tau=0$), the 2-D and 3-D heuristics give identical results.
   The 2-D heuristic always decreases the mean waiting time in contrast to the case of no cooperation (savings between 7% and 25%).

Table 1
Results of the example for variety of λ and τ values

λ	τ	w^{NC}	w^{Min}	$w^{2\text{-D}}$	$\bar{W}_{\text{q}}^{2\text{-D}}$	$\frac{w^{NC}-w^{2\text{-D}}}{w^{NC}}$	$w^{3\text{-D}}$	$\bar{W}_{\text{q}}^{3\text{-D}}$	$\frac{\bar{W}_{\text{q}}^{2\text{-D}}-\bar{W}_{\text{q}}^{3\text{-D}}}{\bar{W}_{\text{q}}^{2\text{-D}}}$
15	0	6	4	5	0.000013	0.166667	5	0.000013	0
15	0.1	6	4	5	0.003104	0.166667	5	0.002661	0.142719
15	0.2	6	4	5	0.011073	0.166667	5	0.009569	0.135826
15	0.4	6	4	5	0.027314	0.166667	5	0.025191	0.077726
15	0.6	6	4	5	0.026561	0.166667	5	0.019583	0.262716
15	1.0	6	4	5	0.03977	0.166667	5	0.037066	0.067991
30	0	7	5	5	0.000278	0.285714	5	0.000278	0
30	0.1	7	5	6	0.005889	0.142857	5	0.048979	—
30	0.2	7	5	6	0.007319	0.142857	6	0.005808	0.206449
30	0.4	7	5	6	0.015187	0.142857	6	0.011651	0.232831
30	0.6	7	5	6	0.018003	0.142857	6	0.015011	0.166195
30	1.0	7	5	6	0.023532	0.142857	6	0.021167	0.100501
50	0	8	6	6	0.000497	0.25	6	0.000497	0
50	0.1	8	6	6	0.045955	0.25	6	0.042088	0.084148
50	0.2	8	6	7	0.009393	0.125	7	0.007992	0.149154
50	0.4	8	6	7	0.015849	0.125	7	0.013249	0.164048
50	0.6	8	6	7	0.019185	0.125	7	0.017145	0.106333
50	1.0	8	6	7	0.023535	0.125	7	0.021668	0.079329
100	0	10	9	9	0.000096	0.1	9	0.000096	0
100	0.1	10	9	9	0.006158	0.1	9	0.005464	0.112699
100	0.2	10	9	9	0.008579	0.1	9	0.008257	0.037534
100	0.4	10	9	9	0.007004	0.1	9	0.006698	0.043689
100	0.6	10	9	9	0.007969	0.1	9	0.007631	0.042414
100	1.0	10	9	9	0.010244	0.1	9	0.009963	0.027431
200	0	15	14	14	0.000001	0.066667	14	0.000001	0
200	0.1	15	14	14	0.000001	0.066667	14	0.000001	0
200	0.2	15	14	14	0.000856	0.066667	14	0.000856	0
200	0.4	15	14	14	0.001682	0.066667	14	0.001682	0
200	0.6	15	14	14	0.002544	0.066667	14	0.002544	0
200	1.0	15	14	14	0.002549	0.066667	14	0.002549	0

Unless $\tau = 0$ and the system is not very congested, the 3-D heuristic improves the mean waiting time significantly (savings between 3% and 26% compared to the 2-D heuristic).

2. On occasion the 3-D heuristic improves the mean manpower obtained with the 2-D heuristic (in the table, this occurred in one of 14 possible cases).

To evaluate the performance of the 3-D heuristic we tested it against two naive policies that human managers might use. We consider the same sets of parameter values as considered above. The following naive rules are considered:

(i) do not allow more than CU customers waiting in the queue per server at the checkout; (ii) do not allow more than S idle servers at the checkout.

We first obtained the following results for the naive policy:

(1) When $CU \geqslant 3,\ S \geqslant 0$ we cannot obtain any feasible solution (impossible to reduce the number of workers required with no cooperation and still satisfy constraints (2) and (3)) for any pair of (λ, τ) values considered.
(2) When $S \geqslant 2, CU \geqslant 0$ we cannot obtain any feasible solution for any pair of (λ, τ) values considered.
(3) With $CU = 0,\ S \geqslant 0$ we can obtain feasible solutions for any pair of (λ, τ) values except when $\tau = 0$ (thus these rules are ignored).
(4) The rule $CU = 1,\ S = 1$ dominates the rule $CU = 1,\ S = 0$ for any (λ, τ) pair.
(5) The rule $CU = 2,\ S = 0$ dominates the rule $CU = 2,\ S = 1$ for any (λ, τ) pair.

In Table 2 we compare the 3-D heuristic against the two naive rules:

(1) Rule $n1$:
$CU = 1,\ S = 1$ (e.g. for $w = 5$ it is equivalent to $\underline{L} = (0, 1, 2, 3, 4)$ $\underline{U} = (2, 4, 6, 8, \infty)$ using 2-D terminology).
(2) Rule $n2$:
$CU = 2,\ S = 0$ (e.g. for $w = 5$ it is equivalent to $\underline{L} = (0, 2, 3, 4, 5)$ $\underline{U} = (3, 6, 9, 12, \infty)$ using 2-D terminology).

The following conclusions can be drawn from Table 2:

(1) Both rules $n1$ and $n2$ are not feasible (impossible to reduce the number of workers required with no cooperation and still satisfy constraints (2) and (3)) for many pairs of (λ, τ) values, whereas the 3-D "optimal policy" provides feasible solutions for all (λ, τ) values.
(2) The 3-D heuristic on two occasions $(\lambda = 30, \tau = 0.1)$ and $(\lambda = 50,\ \tau = 0.1)$ requires one less server than the naive policy.
(3) In all cases where the 3-D heuristic and the naive rules require the same number of servers the mean queue delays of the 3-D are significantly lower (savings over rule $n1$ are between 31% to almost 100% and savings over rule $n2$ are between 37% and almost 100%).

Finally we tested the 2-D and 3-D heuristics for 3 sizes of service facilities—small: $\lambda/\mu = 1.5$, medium: $\lambda/\mu = 4$, and large: $\lambda/\mu = 10$. For each size we ran 16 examples where the other parameters T_0, B_0, γ and τ change. Here are the main findings:

(i) The maximum saving in number of workers of the 3-D heuristic in comparison to the no-cooperation strategy occurs for the middle size facilities: on average 1.9 workers (for both small and large facilities the average is about 1.75 workers).
(ii) The maximum saving in number of workers of the 3-D heuristic in comparison to the 2-D heuristic occurs, for middle size facilities: 0.454 (0.125 for small facilities, and 0 for large facilities).
(iii) For the cases when 3-D heuristic does not provide any savings in the number of workers, maximum relative savings in average waiting time occurs for large facilities: 16.5% (13.7% for small facilities and 12.85% for middle size facilities).

Table 2
Comparison of the 3-D heuristic versus two naive rules

λ	τ	$w^{3\text{-D}}$	$\bar{W}_q^{3\text{-D}}$	w^{n1}	$\bar{W}_q^{n1}$	$\frac{w^{n1}-w^{3\text{-D}}}{w^{n1}}$	$\frac{\bar{W}_q^{n1}-\bar{W}_q^{3\text{-D}}}{\bar{W}_q^{n1}}$	w^{n2}	$\bar{W}_q^{n2}$	$\frac{w^{n2}-w^{3\text{-D}}}{w^{n2}}$	$\frac{\bar{W}_q^{n2}-\bar{W}_q^{3\text{-D}}}{\bar{W}_q^{n2}}$
15	0	5	0.000013	5	0.0139	0	0.999065	5	0.031207	0	0.999583
15	0.1	5	0.002661	5	0.0139	0	0.808561	5	0.031207	0	0.914731
15	0.2	5	0.009569	5	0.0139	0	0.311583	5	0.031207	0	0.69337
15	0.4	5	0.025191	INF	INF			5	0.031207	0	0.192777
15	0.6	5	0.019583	INF	INF			5	0.031207	0	0.372481
15	1.0	5	0.037066	INF	INF			INF	INF		
30	0	5	0.000278	5	0.016551	0	0.983203	5	0.041768	0	0.993344
30	0.1	5	0.048979	6	0.016529	0.166667	NR	6	0.041767	0.166667	NR
30	0.2	6	0.005808	6	0.016529	0	0.648618	6	0.041767	0	0.860676
30	0.4	6	0.011651	INF	INF			INF	INF		
30	0.6	6	0.015011	INF	INF			INF	INF		
30	1.0	6	0.021167	INF	INF			INF	INF		
50	0	6	0.000497	6	0.018843	0	0.973624	6	0.047409	0	0.989517
50	0.1	6	0.042088	7	0.018743	0.142857	NR	7	0.047394	0.142857	NR
50	0.2	7	0.007992	INF	INF			7	0.047394	0	0.831371
50	0.4	7	0.013249	INF	INF			INF	INF		
50	0.6	7	0.017145	INF	INF			INF	INF		
50	1.0	7	0.021668	INF	INF			INF	INF		
100	0	9	0.000096	9	0.022042	0	0.995645	INF	INF		
100	0.1	9	0.005464	INF	INF			INF	INF		
100	0.2	9	0.008257	INF	INF			INF	INF		
100	0.4	9	0.006698	INF	INF			INF	INF		
100	0.6	9	0.007631	INF	INF			INF	INF		
100	1.0	9	0.009963	INF	INF			INF	INF		
200	0	14	0.000001	14	0.02507	0	0.99996	INF	INF		
200	0.1	14	0.000001	INF	INF			INF	INF		
200	0.2	14	0.000856	INF	INF			INF	INF		
200	0.4	14	0.001682	INF	INF			INF	INF		
200	0.6	14	0.002544	INF	INF			INF	INF		
200	1.0	14	0.002549	INF	INF			INF	INF		

INF = Infeasible.
NR = Not relevant.

(iv) The average CPU time (in seconds) for the 2-D and 3-D heuristics are: 1.94, and 4.90 for small facilities, 26.06 and 40.72 for middle-sized facilities, and 1191 and 1336 for large facilities, respectively.

(v) We tried to implement the same types of naive policies discussed above to our size-dependent examples. For middle size facilities we could not find any feasible policy which improves over the no-cooperation policy. For small and large facilities it was not easy to find feasible policies that outperform the no-cooperation policy. For the few examples that worked there was a saving of one server, but with inferior mean waiting time compared to both types 2-D and 3-D heuristics.

We note that for very small examples where it is possible to obtain an optimal solution, the 3-D heuristic very often finds the optimal solution.

7. Management implications

The system we have discussed could be implemented in hardware and software for installation in large retail facilities. The result could potentially be a reduction in manpower in the facility *and* an improvement in customer service levels.

A typical large retail facility could have at any given time say 12 line workers supervised by a store manager or assistant manager located at checkout. Often the manager’s primary job is to adjust worker assignments (in real-time) at checkout, pulling from the back of the store when necessary, to assure that customer queue delays remain at acceptable levels. This job assignment would no longer be required, as the “queue management system” utilizing the methodology of this paper could do it automatically. Additionally, the complement of line workers in the store may be reduced by one or two FTE’s as well, details depending on the parameters of the store. From our discussion in the previous section we may expect to save up to 1.75 workers in a large facility using the 3D heuristic. For a company having a large network of retail facilities, this may result in net reductions of a thousand or more job positions.

Acknowledgements

Applied work on this problem was supported in part by Structured Decisions Corporation (SD-Corp.). of Cambridge, Massachusetts, under contract from the US Postal Service. Methodological work was supported by an NSERC grant and by the Center for Advanced Educational Services at the Massachusetts Institute of Technology. The authors thank Michael Cahn and David Einstein for comments on earlier drafts.

References

[1] Little JDC. A proof of the queueing formula $L = \lambda W$. Operations Research 1961;9:383–7.

[2] Romani J. Un Modelo de la Teoria de Colas con Número Variable de Canales. Trabajos Estadestica 1957;8: 175–89.

[3] Moder JJ, Phillips Jr. CR. Queuing with fixed and variable channels. Operations Research 1962;10:218–31.

[4] Yadin M, Naor P. On queueing systems with variable service capacities. Naval Research Logistics Quarterly 1967;14:43–54.

[5] Serfozo RF. Optimal control of random walks, birth and death processes, and queues. Advanced Applied Probability 1981;13:61–83.

[6] Serfozo RF, Lu FV. M/M/1 Queueing decision processes with monotone hysteretic policies. Operations Research 1984;32:1116–32.

[7] Gross D, Harris CM. Fundamentals of Queueing Theory, 3rd ed. New York: Wiley, 1998.

[8] Berman O, Larson RC. Optimal manpower and switchings in retail services having back room operations and cross-trained workers, MIT Operations Research Center Working Paper, Massachusetts Institute of Technology, Cambridge, MA, September 2000.

Oded Berman is a full professor at the Jospeh L. Rotman School of Management at the University of Toronto. He received his Ph.D. in Operations Research from the Massachusetts Institute of Technology. He has published over 120 articles and has contributed to several books in his field. His main research interests include operations management in the service industry, location theory, network models, and software reliability. He is an Associate Editor for *Operations Research*, Management *Science* and *Transportation Science*, and a member of the editorial boards for *Computers and Operations Research* and the *Journal of Service Research*.

Dick Larson is a professor of Electrical Engineering at MIT and director of MIT's Center for Advanced Educational Services. Over two separate periods totaling 14 years, he served as co-director of MIT's Operations Research Center. He has served in various capacities in our professional organizations, including president of ORSA (1994). He is a member of the National Academy of Engineering. His primary areas of research are OR in the services industries and technology-enabled education. The work on this paper was motivated by a project funded by the US Postal Services to his consulting company Structured Decisions Corporation <http://www.sdcorp.net/>, where Dr. Berman serves as senior technical consultant.